量子鲸鱼优化算法求解作业车间调度问题 *

闫　旭，叶春明，姚远远

(上海理工大学 管理学院, 上海 200093)

摘　要：为了克服基本鲸鱼优化算法（WOA）在解决作业车间调度问题时存在收敛精度低、容易陷入局部最优的缺陷，利用量子计算与优化思想提出了一种量子鲸鱼优化算法（QWOA），并对其进行了计算复杂度分析、全局收敛性证明及仿真实验。通过对 11 个作业车间调度问题基准算例的仿真实验发现，与基本鲸鱼优化算法（WOA）、布谷鸟搜索算法（CS）、灰狼优化算法（GWO）相比，QWOA 算法在最小值、平均值、寻优成功率等方面具有较优结果。研究表明，量子鲸鱼优化算法在解决作业车间调度问题时，具有更高的收敛精度和更好的全局搜索能力，且能够跳出局部最优。

关键词：鲸鱼优化算法；量子计算与优化；作业车间调度；收敛性证明；混合算法

中图分类号：TP301.06

Solving Job-Shop scheduling problem by quantum whale optimization algorithm

Yan Xu, Ye Chunming, Yao Yuanyuan

(*Business School*, *University of Shanghai for Science and Technology*, *Shanghai* 200093, *China*)

Abstract: To overcome whale optimization algorithm's disadvantages of poor convergence and being easily trapped in local optima in solving job-shop scheduling problem, this paper proposed a quantum whale optimization algorithm (QWOA) based on the idea of quantum computation. Then it provided the computational complexity analysis, global convergence proof and simulation experiments of QWOA. Based on a set of 11 JSP benchmark instances, the simulation experiments show that QWOA has better results than whale optimization algorithm (WOA) , cuckoo search (CS) and grey wolf optimizer (GWO) , in the minimum value, average value and success rate. Finally, this paper concludes that QWOA has the merits of higher convergence accuracy, higher local optima avoidance and better exploration.

Key Words: whale optimization algorithm; quantum computing and optimization; job-shop scheduling problem; convergence proof; hybrid algorithm

0　引言

作业车间调度问题（Job-Shop scheduling problem，JSP）是最经典的离散制造系统调度问题。作为一种最困难的组合优化问题，JSP 已经被证明是一种 NP-hard 问题，在多项式时间内找不到一个算法精确地求解其最优解，该问题的求解算法研究已成为车间调度领域的热点[1]。因此，本文将采用新兴的鲸鱼优化算法解决作业车间调度问题。

鲸鱼优化算法（whale optimization algorithm，WOA），由 Mirjalili 等人[2]于 2016 年提出，其仿生学原理源自于座头鲸的狩猎行为。目前，对于鲸鱼优化算法的研究尚处于起步阶段，其正在被逐渐应用到更多的领域；如资源调度领域[3]、选址与路径规划领域[4,5]、工业设计领域[6,7]等。在算法的改进方面，其主要研究目标为对算法的收敛精度和收敛速度的提升，克服算法容易陷入局部最优、收敛速度慢、收敛精度低等方面的不足，具体涉及种群初始化方法的选择与优化、种群多样性的保持与提升、局部开发能力及全局搜索能力的平衡与提升、可行解的空间扰动与遍历等方面；如，Jangir 等人[8]在原有算法的基础上提出了自适应鲸鱼优化算法（AWOA），Ling 等人[9]提出了 LWOA 算法，钟明辉等人[10]提出了一种随机调整控制参数的高效的鲸鱼优化算法（EWOA），刘竹松等人[11]提出了正余混沌双弦鲸鱼优化算法（CSCWOA）。综上所述，鲸鱼优化算法多用于连续优化问题求解，且尚未被应用于解决作业车间调度问题。因此，本文将借鉴量子计算与优化的思想对鲸鱼优化算法进行改进，并将其用于解决离散的作业车间调度问题。

1　作业车间调度问题的数学描述

作业车间调度问题（JSP）一般被描述为：在给定每个工件

基金项目：国家自然科学基金资助项目（71271138）；上海理工大学科技发展项目（16KJFZ028）；上海市高原学科项目（GYXK1201）

作者简介：闫旭（1991-），男，河南商丘人，硕士研究生，主要研究方向为智能计算、组合优化、生产调度等（294669337@qq.com）；叶春明（1964-），男，安徽宣城人，教授，博导，主要研究方向为工业工程、智能计算、生产调度等；姚远远（1989-），女，河南灵宝人，博士研究生，主要研究方向为生产调度、智能算法.

的加工工艺、每个工件使用机器的序列及每个工件每道工序的加工时间的情况下，安排一种工件的加工顺序，使得 n 个待加工的工件在 m 台机器上进行加工，满足某种性能指标最优化。本文所使用的性能指标为“最小化最大完工时间”，且该问题需要满足如下假设及约束：

a）机器不发生故障，从零时刻开始加工工件。

b）同一台机器在同一时刻只能加工一个工件；同一工件同一时刻只能在一台机器上加工。

c）工件的加工工序严格遵守加工时间，一旦开始不得中断；工序间不允许抢占加工。

d）每个工件的工艺路线固定，即工件使用机器的序列给定；每个工件在工艺路线中的每个工序的加工时间固定。但，不同工件的工序间无顺序约束。

通常对作业车间调度问题的数学模型描述有线性规划模型（LP）描述、整数规划模型（IP）描述、析取图模型描述。现利用整数规划模型对其进行如下描述：

$$\min_{\substack{1\le k\le m\\1\le i\le n}} \max\{c_{ik}\} \tag{1}$$

$$s.t.\ c_{ik}-t_{ik}+M(1-a_{ihk})\ge c_{ih} \tag{2}$$

$$c_{jk}-c_{ik}+M(1-x_{ijk})\ge t_{jk} \tag{3}$$

$$c_{ik}\ge 0 \tag{4}$$

$$a_{ihk},x_{ijk}=0,1 \tag{5}$$

$$i,j=1,2,\ldots,n\ ;\ h,k=1,2,\ldots,m \tag{6}$$

其中：t_{ik} 和 c_{ik} 表示工件 i 在机器 k 上的加工时间和完工时间。式（5）中，$a_{ihk}=1$ 表示机器 h 先于机器 k 加工工件 i；$x_{ijk}=1$ 表示工件 i 先于工件 j 在机器 k 上加工；反之，二者皆为 0。M 为一个大数，则式（2）表示工序的前后约束关系；式（3）表示工序的非阻塞约束关系；式（1）为目标函数，表示性能指标最大完工时间的最小化[12]。

2　量子鲸鱼优化算法

2.1　基本鲸鱼优化算法（WOA）

WOA 算法的仿生学原理受启发于座头鲸独特的觅食行为（bubble-net feeding method）。其原理为，座头鲸发现觅食目标后，沿形似螺旋的路径，通过制造气泡网的方式进行觅食，具体行为包含“气泡网攻击（bubble-net attacking method）”、“收缩包围（encircling prey）”、“随机搜寻（Search for prey）”等三种主要方式。其中气泡网攻击捕食猎物的方式又可依据概率分为“螺旋式位置更新（spiral updating position）” 和“缩小包围距离（shrinking encircling mechanism）”两个步骤。则座头鲸的觅食行为可进行如下数学描述：

2.1.1 气泡网攻击

鲸鱼以群体方式觅食，觅食过程中，鲸鱼个体根据随机概率值的大小，对“螺旋式位置更新（spiral updating position:）”和“缩小包围距离（shrinking encircling mechanism）”这两种行为进行选择。其中，“缩小包围距离（shrinking encircling mechanism）”与 2.1.2 节表述相同，“螺旋式位置更新（spiral updating position）”的数学模型可描述为

$$\vec{X}(t+1)=\vec{D'}\cdot e^{bl}\cdot\cos(2\pi l)+\vec{X^*}(t) \tag{7}$$

$$\vec{D'}=\left|\vec{X^*}(t)-\vec{X}(t)\right| \tag{8}$$

其中：“·”表示点乘，“| |”表示取绝对值，t 表示当前的迭代次数，向量 $\vec{X}(t)$ 表示座头鲸的当前位置，向量 $\vec{X}(t+1)$ 表示座头鲸迭代更新后的位置，向量 $\vec{X^*}(t)$ 表示鲸群中的最佳觅食位置，参数 b 为常数用来定义对数螺旋的形状，参数 l 为[-1,1]之间的随机数，两者共同控制鲸鱼个体的螺旋式位置更新方式。当 $l=-1$ 时鲸鱼个体距离觅食目标最近，当 $l=1$ 时鲸鱼个体距离觅食目标最远。

在气泡网攻击过程中，将鲸鱼个体选择位置更新方式的行为概率设为 p。则气泡网攻击（Bubble-net attacking method）过程可以完整描述为

$$\vec{X}(t+1)=\begin{cases}\vec{X^*}(t)-\vec{A}\cdot\vec{D}\ , & p<p^*\\ \vec{D'}\cdot e^{bl}\cdot\cos(2\pi l)+\vec{X^*}(t)\ , & p\ge p^*\end{cases} \tag{9}$$

2.1.2 收缩包围（encircling prey）

根据 2.1.1 节描述，当 $p<p^*$ 时（p^* 取 0.5），鲸鱼个体更新自身在空间中的位置，向较优的位置移动，以实现对猎物的收缩包围。其数学模型描述如下：

$$\vec{D}=\left|\vec{C}\cdot\vec{X^*}(t)-\vec{X}(t)\right| \tag{10}$$

$$\vec{X}(t+1)=\vec{X^*}(t)-\vec{A}\cdot\vec{D} \tag{11}$$

$$\vec{A}=2a\cdot\vec{r_1}-a \tag{12}$$

$$\vec{C}=2\vec{r_2} \tag{13}$$

$$a=2-\frac{2t}{T_{\max}} \tag{14}$$

其中：$T_{\max}$ 为最大迭代次数，a 的值随着迭代次数的增加由 2 向 0 线性递减；$\vec{r_1}$ 和 $\vec{r_2}$ 为（0,1）的随机向量；系数向量 $\vec{C}$ 和 $\vec{A}$ 控制鲸鱼个体的游走方式。需要指出的是，鲸鱼个体采取此种位置更新的前提是 $|\vec{A}|$ 小于 1。即，当 $|\vec{A}|$ 小于 1 时鲸鱼个体 $\vec{X}(t)$ 向鲸群中的当前最优位置 $\vec{X^*}(t)$ 靠近。

2.1.3 随机搜寻（search for prey）

在 2.1.2 节中，式（11）进行位置更新的前提是 $|\vec{A}|$ 小于 1。则当 $|\vec{A}|$ 不小于 1 时，采取如下随机方式进行位置更新，并称之

为随机搜寻（search for prey）。数学模型描述如下：

$$\vec{X}(t+1)=\vec{X}_{rand}-\vec{A}\cdot\vec{D}_{rand} \tag{15}$$

$$\vec{D}_{rand}=\left|\vec{C}\cdot\vec{X}_{rand}-\vec{X}(t)\right| \tag{16}$$

其中：$\vec{X}_{rand}$ 表示鲸鱼群中随机的其他鲸鱼个体。式（15）(16）表示当 $|A|$ 不小于 1 时，鲸鱼个体随机向其他个体游走靠近。

2.2　量子鲸鱼优化算法(QWOA)

2.2.1 量子计算与量子优化

量子计算（quantum computation，QC）由 Feynman 首先于 1982 年提出，现已成为国内外学者关注的前沿学科。在量子计算中，信息的基本存储单位称为量子比特（quantum bit），有别于经典比特，量子比特可以连续、随机地存储于两个极化状态的任意叠加态，按类型可分为单量子比特、双量子比特、多量子比特。其中单量子比特的叠加状态可以表示为

$$|\varphi\rangle=\alpha|0\rangle+\beta|1\rangle \tag{17}$$

$$1=|\alpha|^2+|\beta|^2 \tag{18}$$

其中：$|0\rangle$ 及 $|1\rangle$ 为量子计算中微观粒子的两种基本状态的狄拉克记号表示，α 及 β 表示量子态的概率幅，为一对复数。

量子计算中对量子状态进行逻辑变换的量子装置称为量子逻辑门，是实现量子计算的基础。常用的量子门有“Hadamard 门”“Pauli-X 门”“Pauli-Y 门”“Pauli-Z 门”“相位门”“π/8 门”“量子旋转门”等。其中量子旋转门可表示为

$$R(\theta)=\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix} \tag{19}$$

其量子状态的更新过程可表示为

$$\begin{bmatrix}\alpha'\\ \beta'\end{bmatrix}=R(\theta)\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \tag{20}$$

其中：θ 为旋转角度，$\begin{bmatrix}\alpha'\\ \beta'\end{bmatrix}$ 为更新后的概率幅[13]。

2.2.2 量子鲸鱼优化算法

由于基本鲸鱼优化算法存在收敛精度低、容易陷入局部最优等缺点，因此采用量子计算与量子优化的思想对其进行改进，将量子旋转门操作加入基本鲸鱼优化算法中，对其位置更新方式进行改进，以提升基本算法的种群多样性和收敛精度。已知 $\overline{X}(t)$ 为鲸鱼在搜索空间中的位置，设 $\overline{QX}(t)$ 为鲸鱼经过量子旋转门操作后的位置。

由式（18）可知，$\beta=\sqrt{1-\alpha^2}$，则量子旋转门的更新过程可以简化为

$$\alpha'=\left|\alpha\cdot\cos(\theta)-\sqrt{1-\alpha^2}\cdot\sin(\theta)\right| \tag{21}$$

相对式（20）而言，式（21）的量子旋转门操作方式被称为单链编码量子旋转门操作[14]。鲸鱼个体经过单链编码量子旋转门操作后为

$$\overline{QX}(t)=\left|\overline{X}(t)\cdot\cos(\theta)-\sqrt{1-\overline{X}(t)^2}\cdot\sin(\theta)\right| \tag{22}$$

当鲸鱼个体经过量子旋转门更新后，将其所对应解的质量与旋转之前解的质量进行对比，并选择其中较优者。将量子鲸鱼优化算法简称之为 QWOA(quantum whales optimization algorithm)，其具体步骤可分为：

a)设置迭代次数、种群规模、问题参数。

b)利用随机初始化方法生成初始种群。

c)计算鲸鱼个体的适应度值，找出种群中的当前全局最优解的位置。

d)根据 p 及 $|A|$ 的取值情况对鲸鱼个体的每个维度的变量进行位置更新。

e)对鲸鱼个体各维度的变量进行量子旋转门操作。

f)计算 d)与 e)的更新效果，并对二者进行选择，择优保留。

g)在最大迭代次数内，对鲸鱼群体重复 c)~f)操作。

h)输出求解结果。

至此，量子鲸鱼优化算法的流程可完整描述如图 1 所示。

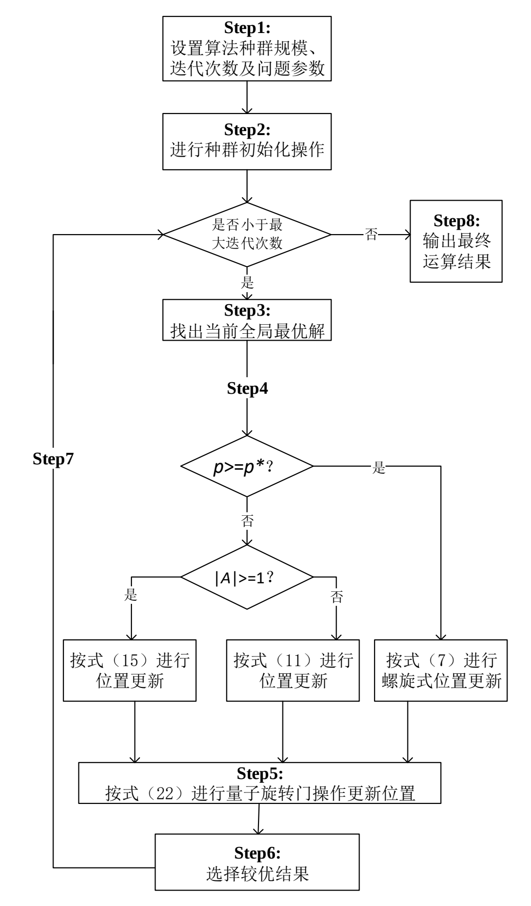


图 1　量子鲸鱼优化算法流程

3　计算复杂度分析及收敛性证明

3.1　计算复杂度分析

利用频度估计的方法对量子鲸鱼优化算法进行计算时间复

杂度和空间复杂分析。设算法的最大迭代次数为 T，种群规模为 POP，变量的维度为 DIM。由 2.2.2 节可知，Step4- Step6 为算法共同的原操作，为处于最内层循环的循环体，语句的执行频度均为 $f(n)=T\cdot POP\cdot DIM$，则算法的渐进时间复杂度 $T(n)=O(f(n))=O(T\cdot POP\cdot DIM)$ 在算法中，可变的储存空间受到种群规模 POP 和变量维度 DIM 的影响，则其空间复杂度可表示为 $S(n)=O(f(n))=O(POP\cdot DIM)$

3.2 算法收敛性证明[15]

利用概率测度法对 QWOA 算法进行全局收敛性证明。根据全局收敛性准则及定理，要证明算法为全局收敛算法，则算法需要满足如下两个假设：

假设 1 $f(D(z,\xi))\le f(z)$，并且如果 $\xi\in S$，则 $f(D(z,\xi))\le f(\xi)$。

其中：f 为最小化问题的目标函数，D 为可以保证产生较优最新解的算子或函数。z 为解空间 R^N 的子集 S 的一个点，能够使得函数的值最小化或在 S 上产生可接受的函数值的下确界。ξ 为算法在样本空间上生成的向量。此假设的意义在于保证优化算法的正确运行，以使算法的解序列收敛于函数 f 在 S 上的下确界，。

假设 2 对于 S 的任意 Borel 子集 A，若其概率测度 $V[A]>0$，则有 $\prod_{k=0}^{\infty}(1-\mu_k[A])=0$。

其中：$\mu_k[A]$ 为由分布 μ_k 产生的对 A 的概率测度。此假设的意义在于，在测度大于 0 的情况下，算法经无穷多次迭代后，不会错过解空间 S 的任意 Borel 子集 A。

按上述两假设要求对 QWOA 算法的全局收敛性进行证明。

引理 1 QWOA 满足假设 1。

证明 1 根据 2.2.2 节描述，可将函数 D 在 QWOA 算法中的描述定义为

$$D(G_t,X_{i,t})=\begin{cases}G_t, & f(g(X_{i,t}))\ge f(G_t)\\ g(X_{i,t}), & f(g(X_{i,t}))<f(G_t)\end{cases}$$

$$D(G_t,X_{i,t})=\begin{cases}G_t, & f(g(X_{i,t}))\ge f(G_t)\\ g(X_{i,t}), & f(g(X_{i,t}))<f(G_t)\end{cases} \tag{23}$$

其中：函数 D 表示对鲸鱼群当前全局最优位置解的计算与选择，连续应用函数 g 对应量子旋转操作函数，$g(X_{i,t})$ 表示鲸鱼个体 i 的第 t 次更新后的位置，G_t 为当前的全局最优解的位置。按照本文算法的定义，可知 G_t 所对应的适应度值是单调不增的，且逐渐向解空间的下确界收敛，因此引理 1 得证。

引理 2 QWOA 满足假设 2。

证明 2 由上述可知：

$$g(X_{i,j,t})=\left|X_{i,j,t}\cdot\cos(\theta)-\sqrt{1-X_{i,j,t}}\cdot\sin(\theta)\right| \tag{24}$$

其中：$X_{i,j,t}$ 为任意迭代步 t，第 i 只鲸鱼个体的第 j 维度变量的位置，$g(X_{i,j,t})$、$X_{i,j,t}$ 的取值范围为[0,1]。则由式（24）可得：

$$\theta=\arccos(\pm g(X_{i,j,t}))-\sigma \tag{25}$$

其中：$\sigma=\arccos(X_{i,j,t})$ 的取值范围为$[0,\frac{\pi}{2}]$，欲使式（25）成立，则 θ 的取值范围为$[\frac{-\pi}{2},\frac{3\pi}{2}]$。设 $X_{i,j,t}$ 为 R^N 第 j 维一点，$g(X_{i,j,t})$ 为 R^N 第 j 维不同于 $X_{i,j,t}$ 的任意点。可知当量子旋转角度落在 θ 范围内时，$X_{i,j,t}$ 到达 $g(X_{i,j,t})$ 的概率大于 0。

定义 $\mu_{i,t}$ 为对应于鲸鱼个体 N 维变量的均匀分布，由上述可知，对 S 的任意 Borel 子集 A，当 $V[A]>0$，可得

$$0<\mu_{i,t}[A]<1 \tag{26}$$

进而可得，由 μ_t 产生的对 A 的概率测度：

$$\mu_t[A]=1-\prod_{i=1}^{S}(1-\mu_{i,t}[A]) \tag{27}$$

通过式（26）（27）得到

$$\prod_{t=0}^{\infty}(1-\mu_t[A])=0 \tag{28}$$

至此，引理 2 得证。

所以，根据证明 1 及证明 2 可证 QWOA 算法满足假设 1 及假设 2，为全局收敛算法。

4 仿真实验与分析

为了确定和检验鲸鱼优化算法解决作业车间调度问题的性能，本文选取作业车间调度标准测试问题库中的 11 个测试问题进行求解并与文献[16]中的其他算法作出比较。算法的运算环境为 Windows10 专业版 32 位操作系统， Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz 处理器，4G RAM。算法的编程实现软件为 Matlab2015b。

4.1 WOA 与灰狼优化算法和布谷鸟搜索算法的比较

灰狼优化算法[17]（grey wolf optimizer，GWO）由 Mirjalili 等人于 2014 年提出；布谷鸟搜索算法[18]（cuckoo search，CS）由 Yang 等于 2009 年提出；二者皆为新兴的智能算法。为了更准确地与 GWO 和 CS 算法就 JSP 问题进行比较，将 WOA 算法在求解 JSP 问题时的参数设置与文献[16]保持一致，即种群规模为 30，最大迭代次数为 500，算法独立运行次数为 30。得到如表 1 所示结果。

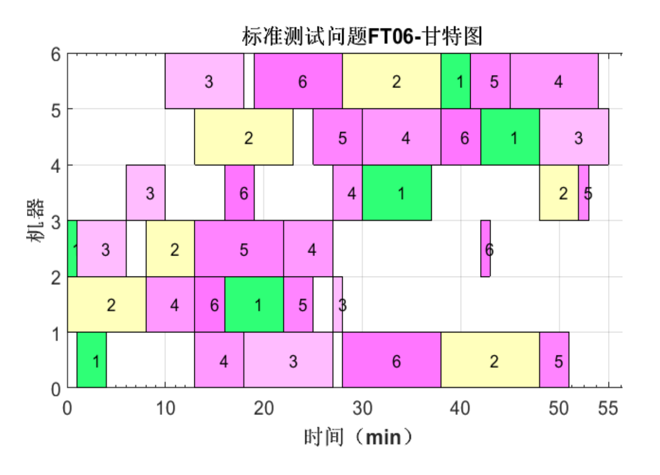
由表 1 的结果可知，在种群规模、迭代次数及算法运算次数相同的条件下，WOA 算法与 GWO 算和 CS 算法在 FT06、

LA01、LA06、LA11 等问题中都能收敛到当前已知最优解（部分甘特图如图 2 所示），但在寻优成功率上 WOA 算法要劣于 GWO 算法和 CS 算法。

在最小值方面，WOA 算法优于 CS 算法，但劣于 GWO 算法。在平均值方面，WOA 算法劣于 GWO 算法，少部分优于 CS 算法。

表 1 基本鲸鱼优化算法与灰狼优化算法和布谷鸟搜索算法的比较

实例名称（问题规模）	已知最优	WOA			GWO			CS		
		最小值	平均值	寻优成功率	最小值	平均值	寻优成功率	最小值	平均值	寻优成功率
FT06（6×6）	**55**	**55**	57.8	**16.7%**	**55**	57.6	20%	**55**	55.0	**96.7%**
FT10（10×10）	930	1065	1155.2	0%	989	1061.8	0%	1075	1131.6	0%
FT20（20×5）	1165	1334	1467	0%	1274	1378.4	0%	1368	1429.2	0%
LA01（10×5）	**666**	**666**	689.8	**30%**	**666**	673.7	56.7%	**666**	667.0	**86.7%**
LA06（15×5）	**926**	**926**	933.8	**53.3%**	**926**	926.6	96.7%	**926**	926	**100%**
LA11（20×5）	**1222**	**1222**	1241.0	**40%**	**1222**	1222.1	93.3%	**1222**	1226.8	**56.7%**
LA16（10×10）	945	994	1080.9	0%	979	1026.4	0%	1015	1043.8	0%
LA21（15×10）	1046	1256	1337.6	0%	1167	1235.3	0%	1267	1331.5	0%
LA26（20×10）	1218	1510	1598.7	0%	1345	1459.3	0%	1487	1614.6	0%
LA31（30×10）	1784	1978	2161.1	0%	1871	1979.8	0%	2102	2198.8	0%
LA36（15×15）	1268	1555	1658.5	0%	1393	1501.7	0%	1600	1655	0%



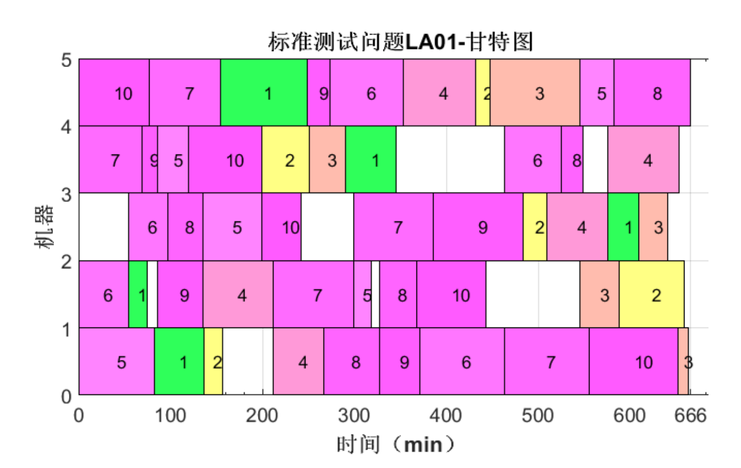


图 2 部分标准测试问题的甘特图

上述结果说明，WOA 算法虽然能够成功运用于在作业车间调度问题中，但与 GWO 与 CS 算法相比较并无明显优势，仍然存在较大改进空间。因此，本文采用改进后的 QWOA 算法继续对上述问题进行求解。

4.2 QWOA 与 WOA 的比较

利用量子鲸鱼优化算法对上述问题进行求解，将算法的种群规模设置为 30，最大迭代次数设置为 500，量子旋转的角度设置为 rand*2*π 。运算结果的对比如表 2 所示。

表 2 中，最小值提升、平均值提升、寻优成功率提升的计算方法分别为：（WOA 最小值-QWOA 最小值）/ WOA 最小值、（WOA 平均值-QWOA 平均值）/WOA 平均值、QWOA 寻优成功率-WOA 寻优成功率。

通过 QWOA 与 WOA 算法就求解作业车间调度问题的比较可知，量子鲸鱼优化算法对算法的寻优成功率方面有很大的提升，在 FT06、LA01、LA06、LA11 等问题上的寻优成功率的值均在 83%以上，其中 LA01、LA06、LA11 的寻优成功率分别高达 93.3%、100%、100%，寻优成功率的平均提升率为 59.2%；且在 LA31 问题中，收敛到了 WOA、GWO、CS 算法所无法收敛到的已知最优解，同时在 LA16 问题中也近乎收敛到已知最优解。

而且，QWOA 对最小值及平均值的提升效果也十分显著，取值明显优于 WOA、GWO、CS 算法，对 WOA 算法的最小值及平均值的平均提升率分别为 10.5%、9.7%。

同时，图 3 中 QWOA 与 WOA、CS、GWO 等算法寻优曲线的特征对比，说明量子鲸鱼优化算法在求解作业车间调度问题的最优解时，收敛到最优值所需要的迭代次数更少;收敛的精度更高；跳出局部最优进行全局搜索的能力更强。

结合算法寻优的方差情况（受篇幅限制未予展示），总体而言，利用量子计算与量子优化的思想对基本鲸鱼优化算法进行改进后，量子鲸鱼优化算法在求解作业车间调度优化问题时具

有更高的收敛精度、更少的收敛次数和更强的全局搜索能力和鲁棒性。

表 2　量子鲸鱼优化算法与基本鲸鱼优化算法的比较

实例名称（问题规模）	已知最优	QWOA			WOA			改进提升效果		
		最小值	平均值	寻优成功率	最小值	平均值	寻优成功率	最小值	平均值	寻优成功率
FT06（6×6）	**55**	**55**	**55.3**	**83.3%**	**55**	57.8	**16.7%**	0%	**4.3%**	**66.6%**
FT10（10×10）	930	966	1007.2	0%	1065	1155.2	0%	**9.3%**	**12.8%**	0.0%
FT20（20×5）	1165	1207	1252.1	0%	1334	1467	0%	**9.5%**	**14.6%**	0.0%
LA01（10×5）	**666**	**666**	**667.5**	**93.3%**	**666**	689.8	**30%**	0%	**3.2%**	**63.3%**
LA06（15×5）	**926**	**926**	**926**	**100%**	**926**	933.8	**53.3%**	0%	**0.8%**	**46.7%**
LA11（20×5）	**1222**	**1222**	**1222**	**100%**	**1222**	1241.0	**40%**	0%	**1.5%**	**60.0%**
LA16（10×10）	945	946	994.3	0%	994	1080.9	0%	**4.8%**	**8.0%**	0.0%
LA21（15×10）	1046	1110	1146.2	0%	1256	1337.7	0%	**11.6%**	**14.3%**	0.0%
LA26（20×10）	1218	1269	1331.1	0%	1510	1598.7	0%	**16%**	**16.7%**	0.0%
LA31（30×10）	**1784**	**1784**	**1846.2**	**6.7%**	1978	2161.1	0%	**9.8%**	**14.6%**	**6.7%**
LA36（15×15）	1268	1357	1403.8	0%	1555	1658.5	0%	**12.7%**	**15.4%**	0.0%

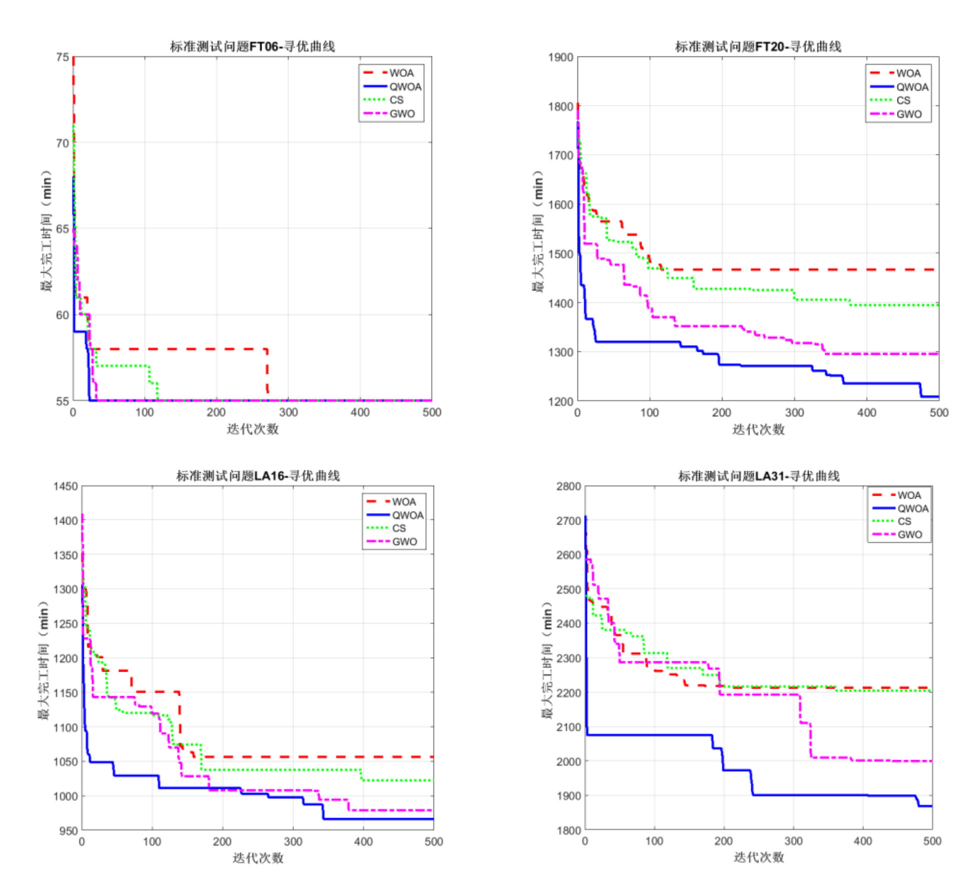


图 3　部分标准测试问题的寻优曲线对比

5　结束语

鲸鱼优化算法（WOA）是新兴的群智能算法，多应用于连续优化问题，在作业车间调度（JSP）这类离散组合优化问题中较少应用。本文验证了 WOA 算法解决 JSP 问题的可行性，并发现了其在解决 JSP 问题时的缺陷。为了克服 WOA 算法解决 JSP 问题时的缺陷，本文提出了一种量子鲸鱼优化算法（QWOA），并对其进行了计算复杂度分析、收敛性证明及实验仿真分析。

通过对比分析发现，本文所提出的 QWOA 算法在解决 JSP 问题时，具有迭代次数少、收敛精度高、全局搜索能力强等优点。

在未来的工作中，本文将利用 WOA 算法对多目标组合优化问题进行研究，并进一步提升 WOA 算法的应用性能。

参考文献：

[1] Cheng T C E, Peng B, Lü Z. A hybrid evolutionary algorithm to solve the job shop scheduling problem [J]. Annals of Operations Research, 2016, 242 (2): 223-237.

[2] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[3] Trivedi I N, Bhoye M, Bhesdadiya R H, et al. An emission constraint environment dispatch problem solution with microgrid using whale optimization algorithm [C]// Proc of Power Systems Conference. 2017.

[4] Prakash D B, Lakshminarayana C. Optimal siting of capacitors in radial distribution network using whale optimization algorithm [J]. Alexandria Engineering Journal, 2016.

[5] Dao T K, Pan T S, Pan J S. A multi-objective optimal mobile robot path planning based on whale optimization algorithm [C]// Proc of IEEE International Conference on Signal Processing. 2017: 337-342.

[6] Kaveh A, Ghazaan M I. Enhanced whale optimization algorithm for sizing optimization of skeletal structures [J]. Mechanics Based Design of Structures & Machines, 2017, 45: 345-362.

[7] Oliva D, Aziz M A E, Hassanien A E. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm [J]. Applied Energy, 2017, 200: 141-154.

[8] Jangir P, Trivedi I N, Jangir N, et al. A novel adaptive whale optimization algorithm for global optimization [J]. Indian Journal of Science & Technology, 2016, 9 (38) .

[9] Ling Y, Zhou Y, Luo Q. Lévy Flight trajectory-based whale optimization algorithm for global optimization [J]. IEEE Access, 2017, 5 (99): 6168-6186.

[10] 钟明辉，龙文. 一种随机调整控制参数的鲸鱼优化算法 [J]. 科学技术与工程, 2017, 17 (12): 68-73.

[11] 刘竹松，李生. 正余混沌双弦鲸鱼优化算法 [J]. 计算机工程与应用, 2017.

[12] Cheng T C E, Peng B, Lü Z. A hybrid evolutionary algorithm to solve the job shop scheduling problem [J]. Annals of Operations Research, 2016, 242 (2): 223-237.

[13] Dey S, Saha I, Bhattacharyya S, et al. Multi-level thresholding using quantum inspired meta-heuristics [J]. Knowledge-Based Systems, 2014, 67 (3): 373-400.

[14] 高洪元，刁鸣. 量子群智能及其在通信技术中的应用 [M]. 北京：电子工业出版社, 2016.

[15] Fang W, Sun J, Chen H, et al. A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population [J]. Information Sciences, 2016, 330 (C): 19-48.

[16] 姚远远，叶春明. 求解作业车间调度问题的改进混合灰狼优化算法 [J/OL]. 计算机应用研究, 2018, 35 (5) (2017-06-14) . http://kns. cnki. net/kcms/detail/51. 1196. TP. 20170614. 1317. 050. html.

[17] Mirjalili S. How effective is the Grey Wolf optimizer in training multi-layer perceptrons [J]. Applied Intelligence, 2015, 43 (1): 150-161.

[18] Li X, Yin M. Modified cuckoo search algorithm with self adaptive parameter method [J]. Information Sciences, 2015, 298 (C): 80-97.